CONCETTI BASE DELLA PROGRAMMAZIONE

TIPI DI VARIABILI

In programmazione ci sono diversi tipi di variabili, ognuno dei quali è utilizzato per immagazzinare un diverso tipo di dato. Di seguito l'elenco dei principali tipi di variabili:

1. Integer: una variabile intera viene utilizzata per immagazzinare numeri interi. In C# ad esempio, una variabile intera viene dichiarata con la keyword "**int**" seguita dal nome della variabile: `int numero = 5;`
2. Double: una variabile double viene utilizzata per immagazzinare numeri decimali. In C# ad esempio, una variabile double viene dichiarata con la keyword "**double**" seguita dal nome della variabile: `double numero = 3.14;`
3. Float: una variabile float viene utilizzata per immagazzinare numeri decimali con precisione singola. In C# ad esempio, una variabile float viene dichiarata con la keyword "**float**" seguita dal nome della variabile: `float numero = 3.14f;`

   La differenza principale tra float e double è la precisione: float utilizza 32 bit per la rappresentazione del numero, mentre double ne utilizza 64. Pertanto, float è più veloce ma meno preciso di double.
4. String: una variabile stringa viene utilizzata per immagazzinare testo. In C# ad esempio, una variabile stringa viene dichiarata con la keyword "**string**" seguita dal nome della variabile: `string nome = "Mario";`

5. Boolean: una variabile booleana viene utilizzata per immagazzinare un valore di verità, ovvero true o false. In C# ad esempio, una variabile booleana viene dichiarata con la keyword "**bool**" seguita dal nome della variabile: `bool risultato = true;`
6. Char: una variabile carattere viene utilizzata per immagazzinare un singolo carattere. In C# ad esempio, una variabile carattere viene dichiarata con la keyword "**char**" seguita dal nome della variabile: `char carattere = 'a';`
7. Varchar: una variabile varchar viene utilizzata per immagazzinare stringhe di lunghezza variabile. In SQL ad esempio, una variabile varchar viene dichiarata con la keyword "**varchar**" seguita dalla lunghezza massima della stringa e dal nome della variabile: `varchar(50) nome = 'Mario';`. La lunghezza massima di una variabile varchar può variare a seconda del database utilizzato.
8. Array: un array è una collezione di variabili dello stesso tipo. In C# ad esempio, un array viene dichiarato con il tipo di variabile seguito dalle parentesi quadre e dal nome della variabile: `int[] numeri = {1, 2, 3, 4};`
9. Object: una variabile di tipo object può contenere qualsiasi tipo di dato. In C# ad esempio, una variabile di tipo object viene dichiarata con la keyword "**object**" seguita dal nome della variabile: `object variabile = "testo";`
10. Enum: un'enumerazione viene utilizzata per creare un set di costanti denominate. In C# ad esempio, un'enumerazione viene dichiarata con la keyword "**enum**" seguita dal nome della variabile e dalle costanti denominate:

```
enum GiorniSettimana
{
    Lunedi,
    Martedi,
    Mercoledi,
    Giovedi,
    Venerdi,
    Sabato,
    Domenica
};
```

11. Long: una variabile long viene utilizzata per immagazzinare numeri interi molto grandi. In C# ad esempio, una variabile long viene dichiarata con la keyword "**long**" seguita dal nome della variabile: `long numero = 1000000000;`

12. Short: una variabile short viene utilizzata per immagazzinare numeri interi più piccoli rispetto ad int. In C# ad esempio, una variabile short viene dichiarata con la keyword "**short**" seguita dal nome della variabile: `short numero = 10;`
13. Byte: una variabile byte viene utilizzata per immagazzinare numeri interi positivi compresi tra 0 e 255. In C# ad esempio, una variabile byte viene dichiarata con la keyword "**byte**" seguita dal nome della variabile: `byte valore = 200;`
14. Decimal: una variabile decimal viene utilizzata per immagazzinare numeri decimali con una precisione maggiore rispetto a double. In C# ad esempio, una variabile decimal viene dichiarata con la keyword "**decimal**" seguita dal nome della variabile: `decimal numero = 3.14m;`
15. DateTime: una variabile DateTime viene utilizzata per immagazzinare data e ora. In C# ad esempio, una variabile DateTime viene dichiarata con la keyword "**DateTime**" seguita dal nome della variabile: `DateTime data = new DateTime(2022, 05, 05);`
16. TimeSpan: una variabile TimeSpan viene utilizzata per immagazzinare una durata di tempo. In C# ad esempio, una variabile TimeSpan viene dichiarata con la keyword "**TimeSpan**" seguita dal nome della variabile: `TimeSpan durata = new TimeSpan(2, 30, 0);`

CONTROLLI CONDIZIONALI

I controlli condizionali sono uno strumento fondamentale nella programmazione e sono utilizzati per eseguire un blocco di codice solo se una determinata condizione è soddisfatta. I principali controlli condizionali utilizzati nella programmazione sono i seguenti:

1. **if**: Il controllo if viene utilizzato per eseguire un blocco di codice solo se una determinata condizione è soddisfatta. La sintassi di base è la seguente:

   ```
   if (condizione) {
     // blocco di codice da eseguire
   }
   ```

   Ad esempio, il seguente codice esegue un blocco di codice solo se il valore della variabile `numero` è maggiore di 10:

```
int numero = 15;
if (numero > 10) {
  // blocco di codice da eseguire
}
```

2. **if-else**: Il controllo if-else viene utilizzato per eseguire un blocco di codice se una determinata condizione è soddisfatta, altrimenti viene eseguito un altro blocco di codice. La sintassi di base è la seguente:

```
if (condizione) {
  // blocco di codice da eseguire se la condizione è soddisfatta
} else {
  // blocco di codice da eseguire se la condizione non è soddisfatta
}
```

   Ad esempio, il seguente codice esegue un blocco di codice se il valore della variabile `numero` è maggiore di 10, altrimenti esegue un altro blocco di codice:

```
int numero = 15;
if (numero > 10) {
  // blocco di codice da eseguire se il numero è maggiore di 10
} else {
  // blocco di codice da eseguire se il numero non è maggiore di 10
}
```

3. **switch**: Il controllo switch viene utilizzato per eseguire un blocco di codice diverso in base al valore di una variabile. La sintassi di base è la seguente:

```
switch (variabile) {
  case valore1:
    // blocco di codice da eseguire se la variabile ha il valore valore1
    break;
  case valore2:
    // blocco di codice da eseguire se la variabile ha il valore valore2
    break;
  // altri casi possibili
  default:
    // blocco di codice da eseguire se nessuno dei casi precedenti è soddisfatto
    break;
}
```

Ad esempio, il seguente codice esegue un blocco di codice diverso in base al valore della variabile `giorno`:

```
string giorno = "martedì";
switch (giorno) {
  case "lunedì":
    // blocco di codice da eseguire se il giorno è lunedì
    break;
  case "martedì":
    // blocco di codice da eseguire se il giorno è martedì
    break;
  // altri casi possibili
  default:
    // blocco di codice da eseguire se nessuno dei casi precedenti è soddisfatto
    break;
}
```

CICLI

I cicli sono uno strumento fondamentale nella programmazione e sono utilizzati per ripetere un blocco di codice un certo numero di volte o finché una condizione è soddisfatta. I principali cicli utilizzati nella programmazione sono i seguenti:

1. Ciclo **for**: Il ciclo for viene utilizzato per eseguire un blocco di codice un certo numero di volte. La sintassi di base è la seguente:

   ```
   for (inizializzazione; condizione; incremento/decremento) {
     // blocco di codice da eseguire
   }
   ```

   Ad esempio, il seguente codice esegue un blocco di codice 5 volte:

   ```
   for (int i = 0; i < 5; i++) {
     // blocco di codice da eseguire
   }
   ```

2. Ciclo **while**: Il ciclo while viene utilizzato per eseguire un blocco di codice finché una determinata condizione è soddisfatta. La sintassi di base è la seguente:

```
while (condizione) {
  // blocco di codice da eseguire
}
```

Ad esempio, il seguente codice esegue un blocco di codice finché il valore della variabile `i` è minore di 10:

```
int i = 0;
while (i < 10) {
  // blocco di codice da eseguire
  i++;
}
```

3. Ciclo **do-while**: Il ciclo do-while viene utilizzato per eseguire un blocco di codice almeno una volta, e poi ripetere il ciclo finché una determinata condizione è soddisfatta. La sintassi di base è la seguente:

```
do {
  // blocco di codice da eseguire
} while (condizione);
```

Ad esempio, il seguente codice esegue un blocco di codice almeno una volta e poi ripete il ciclo finché il valore della variabile `i` è minore di 10:

```
int i = 0;
do {
  // blocco di codice da eseguire
  i++;
} while (i < 10);
```

4. Ciclo **foreach**: Il ciclo foreach viene utilizzato per eseguire un blocco di codice per ogni elemento di una collezione. La sintassi di base è la seguente:

```
foreach (tipo elemento in collezione) {
  // blocco di codice da eseguire
}
```

Ad esempio, il seguente codice esegue un blocco di codice per ogni elemento della lista `numeri`:

```
List<int> numeri = new List<int> { 1, 2, 3, 4, 5 };
foreach (int numero in numeri) {
  // blocco di codice da eseguire per ogni numero della lista
}
```